CLAIMS

What is claimed is:

1. A method for processing HDLC data, the method comprising:

retrieving previously stored first state information for a first HDLC channel;

using the retrieved first state information to resume processing of the first HDLC channel;

writing second state information for the first HDLC channel to storage;

retrieving previously stored third state information for a second HDLC channel; and

using the retrieved third state information to resume processing of the second HDLC channel.

2. The method of claim 1, further comprising:

writing fourth state information for the second HDLC channel to storage;

retrieving the second state information; and

using the second state information to resume processing of the first HDLC channel.

3. The method of claim 1, in which the state information comprises: a frame status bit, a current frame check sequence value, and a count of the number of consecutive ones received in the currently received frame.

4. The method of claim 3, in which the state information further comprises a count of the amount of data needed before a currently received frame will be long enough to be valid.

5. The method of claim 3, in which the state information further comprises an indication of whether the last frame was terminated by a flag.

6. The method of claim 1, in which the state information comprises a byte residue.

7. The method of claim 6, in which the state information further comprises a count of the number of valid bits in the byte residue.

8. The method of claim 1, performed by an HDLC coprocessor of a network processing engine.

9. The method of claim 1, in which the processing of the first and second channels comprises one of: encapsulating raw data into HDLC encoded data, or (b) de-encapsulating HDLC encoded data into raw data.

10. A network processing engine comprising:

a processor;

memory;

computer code stored in said memory, which, when executed by the processor, is operable to cause the processor to perform actions comprising:

retrieving previously stored first state information for a first HDLC channel;

using the retrieved first state information to resume processing of the first HDLC channel;

writing second state information for the first HDLC channel to memory;

retrieving previously stored third state information for a second HDLC channel; and

using the retrieved third state information to resume processing of the second HDLC channel.

11. The network processing engine of claim 10, in which the state information comprises a residue of bits that fall outside a predefined byte boundary.

12. The network processing engine of claim 11, in which the state information further comprises a count of the number of valid bits in the residue.

13. The network processing engine of claim 10, in which the processor comprises an HDLC coprocessor, and the memory comprises memory contained within the HDLC coprocessor.

14. A computer program package embodied on a computer readable medium, the computer program package comprising instructions that, when executed by a processor, cause the processor to perform actions comprising:

retrieving stored first state information for a first HDLC channel;

using the retrieved first state information to resume processing of the first HDLC channel;

writing second state information for the first HDLC channel to storage;

retrieving previously stored third state information for a second HDLC channel; and

using the retrieved third state information to resume processing of the second HDLC channel.

15. The computer program package of claim 14, further comprising instructions that cause the processor to perform actions comprising:

writing fourth state information for the second HDLC channel to storage;

retrieving the second state information; and

using the second state information to resume processing of the first HDLC channel.

16. The computer program package of claim 14, in which the computer readable medium comprises a memory unit in a network processing engine, and in which the processor comprises an HDLC coprocessor in said network processing engine.

17. The computer program package of claim 14, in which the state information includes channel context information comprising:

a frame status bit, a current frame check sequence value, and a count of the number of consecutive ones received in the current frame.

18. The computer program package of claim 14, in which the state information comprises:

a residue of bits that fall outside a predefined byte boundary, and

a count of the number of bits in the residue.

19. A network processing engine configured to encapsulate and de-encapsulate multiple HDLC channels at a time.

20. The network processing engine of claim 19, comprising an HDLC coprocessor for locally storing state information for each HDLC channel being processed, and for retrieving the stored state information for a given HDLC channel and using the retrieved state information to resume processing of the given HDLC channel.

21. A system comprising:

a network processing engine, the network processing engine being configured to:

obtain multiple channels of HDLC data, the multiple channels of HDLC data being time-division multiplexed together;

accumulate chunks of HDLC data from each channel; and

pass the chunks of HDLC data to an HDLC coprocessor;

an HDLC coprocessor, the HDLC coprocessor being configured to:

obtain the chunks of HDLC data for each channel from said network processing engine; and de-encapsulate the chunks of HDLC data.

22. The system of claim 21, in which the HDLC coprocessor is further configured to:

obtain raw data;

encapsulate the raw data into HDLC frames; and

pass the encapsulated data to a component of the network processing engine for transmission over a network.